Project Euler - Problem 4 - Largest Palindrome Product

A palindromic number reads the same both ways. The largest palindrome made from the product of two 2-digit numbers is 9009 = 91 × 99.

Find the largest palindrome made from the product of two 3-digit numbers.

In [5]:
```
import time

start = time.time()

import numpy as np
seq = np.arange(100, 1000, 1)[np.newaxis]
matrix_prod = seq*seq.T
answer_vector = []

for i in range(0, len(matrix_prod)-1, 1):
    for j in range(0, len(matrix_prod)-1, 1):
        if len(str(matrix_prod[i,j])) == 5:
            if str(matrix_prod[i,j])[0] == str(matrix_prod[i,j])[4] and str(matrix_prod[i,j])[1] == str(matrix_prod[
i,j])[3]:
                answer_vector.append(matrix_prod[i,j])
            else:
                pass
        else:
            if str(matrix_prod[i,j])[0] == str(matrix_prod[i,j])[5] and str(matrix_prod[i,j])[1] == str(matrix_prod[
i,j])[4] and str(matrix_prod[i,j])[2] == str(matrix_prod[i,j])[3]:
                answer_vector.append(matrix_prod[i,j])
            else:
                pass

# Print first 40 elements
print(answer_vector[1:40])


# End
end = time.time()
```

```
[11211, 12221, 13231, 14241, 15251, 16261, 17271, 18281, 19291, 20402, 21412, 22422, 23432, 24442, 25452, 26462, 2747
2, 28482, 29492, 30603, 31613, 32623, 33633, 34643, 35653, 36663, 37673, 38683, 39693, 40804, 41814, 42824, 43834, 44
844, 45854, 46864, 47874, 48884, 49894]
```

In [6]:
```
# Maximum value
print("The max value is: " + str(max(answer_vector)))

# Running time
print("time: " + str(end - start))
```

```
The max value is: 906609
time: 2.361721992492676
```

Since the the product is either a five-digit or a six-digit number, we need to break into two cases.

The algorithm takes each element of the matrix and evaluate if it satisfies the conditons of a palindromic number based on its length. If it does, then the number is added on the answer_vector. Otherwise, the algorithm moves to the next number.

In [ ]:
```

```